Statistical Properties of a Class of Randomized Binary Search Algorithms

Ye Xia[*] Shigang Chen[†]

Department of Computer and Information Science and Engineering

University of Florida

Abstract

In this paper, we analyze the statistical properties of a randomized binary search algorithm and its variants. The basic discrete version of the problem is as follows. Suppose there are total m items, numbered 1, 2, ..., m, out of which the first k items are marked, where k is unknown. The objective is to choose one of the marked items uniformly at random. In each step of the basic algorithm, a number y is chosen uniformly at random from 1 to x, where x is the number chosen in the previous step and is equal to m in the first step. A query is made about y. If y is marked, the algorithm returns. We will also consider batch versions of this algorithm in which multiple numbers are chosen in each step and multiple queries are made in parallel. We give the mean and variance (exact or asymptotic) for the number of search steps in each version of the algorithm, and when possible, we give its distribution. We also analyze the access or hit pattern to the entire search space. The basic algorithm is fairly efficient in terms of the number of search steps, and also has small variance. The two batch versions of the algorithm can be used separately or combined to further reduce the number search steps and its variance.

[*]Email: yx1@cise.ufl.edu.

[†]Email:sgchen@cise.ufl.edu

keywords Analysis of algorithms, parallel and distributed algorithms, randomized algorithms, randomized binary search, distributed search, load balancing

Classification 68W10, 68W15, 68W20, 68W40, 60G46, 60G55

1 Introduction

In this paper, we analyze the statistical properties of a randomized binary search algorithm and its variants. The basic discrete-version of the problem is as follows. Suppose there are total m items, numbered 1, 2, ..., m, out of which the first k items are marked, where k is unknown. The objective is to choose one of the marked items uniformly at random.

This problem arises in caching and load balancing in distributed environment, such as the peer-to-peer network. Suppose the computers in the network are numbered from 1 to m, and suppose a particular popular file is cached at servers 1 to k, where $1 \leq k \leq m$. Due to the fluctuating demand for the file and due to the distributed nature of the scenario, the clients who wish to request the file do not necessarily know the number k. It may be undesirable to use designated centralized servers for keeping such information, because either the participants of the peer-to-peer network are not willing to act as servers, or maintenance of the servers and their information content is highly complex in an extremely dynamic network environment where the participants join and leave frequently. An algorithm that solves the aforementioned problem allows us to choose one of the cache servers for downloading the file uniformly at random, and hence, balance the load to the cache servers 1 to k.

As a remark, the assumption that the first k servers are always marked, i.e., contain the file, can be realized in practice by either actively moving files down to the first k servers through a background protocol, or by adding another level of indirection with hash functions. In the latter case, suppose there are total m hash functions, numbered from 1 to m. When a hash function is applied to a file or file name, a server ID will be returned. Depending on the request rate for the file, the number of hash functions that are actually used, denoted by k, will vary. One can design a policy for hash function usage so that, when the total request rate warrants the use of k hash functions, only hash functions 1 to k are actually used. A background hash function management protocol is needed to enforce such a policy.

One strategy is to determine the unknown k by (deterministic) binary search, asking if the current item is marked or not at each step. Then, we choose i from 1 to k uniformly at random. In the distributed caching application, a query message is sent to the computer being queried during each binary search step, checking whether the computer is marked, i.e., whether it contains a copy of the file. In this strategy, the eventual choice of the cache server is uniformly random, but the query-message load to all the computers is not uniform. All query messages traverse the same small set of computers. We call a query to each computer an *access* or a *hit* to the computer, which, in our algorithms, corresponds to an inquiry whether a number is marked.

We will consider a randomized binary search algorithm, which accesses the marked items (corresponding to the cache servers) uniformly. In each step, a number y is chosen uniformly at random from 1 to x, where x is the number chosen in the previous step and is equal to m in the first step. If y is marked, the algorithm returns y; otherwise, the search continues. Being a variation of the deterministic binary search algorithm, it is expected that the mean number of search steps is $O(\log \frac{m}{k})$.

Our first contribution by this paper is that we can give more precise expression for the mean number of search steps than its order of growth (i.e., $O(\log \frac{m}{k})$). For instance, we will give the constant factor C in $C \log \frac{m}{k}$. This is important for the intended application of distributed search in the peer-to-peer network. Suppose the network delay between a pair of peers is 50 ms and each search message traverses 10 peers. Then, each search step takes 0.5 seconds to complete. If the randomized binary search algorithm takes 24 search steps on average, the total search time will be 12 seconds, which is long for interactive applications. Reducing the search time by a factor, say 8, can be significant.

Our second contribution is that we can describe the statistical fluctuation in the number of search steps by computing the variance, and when possible, the distribution. Our third contribution is related to the fact that, even though the algorithm returns one of the k marked items uniformly at random, the number of hits to each location from 1 to m is not uniform. In the terminology of the caching/load-balancing application, the query traffic seen by different computers is not uniform. We have derived the access pattern for the un-marked items.

As our fourth contribution, we have analyzed two batch versions of the randomized binary search algorithm that can reduce the number of search steps by running s queries in parallel. In the first version, presented in Section 4, the target numbers of the parallel queries are $y_1 \in [0, y_0]$, $y_2 \in [0, y_1]$, ..., $y_s \in [0, y_{s-1}]$[1], where each y_i, $1 \leq i \leq s$, is chosen uniformly at random within the respective interval and y_0 is the smallest target number chosen in the previous search step. The algorithm returns the first y_i landing in the marked region. It is still true that the number returned by the algorithm is chosen uniformly at random from the marked region, and the number of search steps is cut down by a factor of s. The access pattern to the un-marked region remains the same as in the basic algorithm. However, the access pattern to the marked region is no longer uniform. We give an expression for the access pattern, which yields the practical limit of this algorithm. In the second batch algorithm, presented in Section 5, the target numbers of the s parallel queries are all chosen uniformly at random on $[0, x]$, where x is the smallest target number chosen in the previous step. This algorithm has the properties that the number returned by the algorithm is chosen uniformly at random from the marked region, the access pattern to the marked region is uniform, and that, the access pattern to the un-marked region remains the same as in the basic algorithm except that the magnitude increases by s times. However, the mean number of search steps is cut down by a factor of $\ln s$ only, when compared with the basic algorithm. We have derived fairly precise expressions for the mean and variance of the number of search steps. The techniques required for analyzing the second batch algorithm is less conventional, and will be the focus of this paper.

[1] It will become clear shortly that, for ease of analysis, we use a continuous approximation to the discrete algorithm. The entire search space is the interval $[0, 1]$ and the marked region is $[0, a]$.

2 Discrete Case - Algorithm 1

Suppose there are total m items, numbered 1, 2, ..., m, out of which the first k items are marked, where k is unknown. The objective is to select one of the marked items uniformly at random. We consider the following randomized binary search algorithm, known as Algorithm 1. First, initialize $x \Leftarrow m$. In each step, a number y is chosen uniformly at random from 1 to x. If y is marked, the algorithm returns y; otherwise, the search continues with $x \Leftarrow y$.

It is easy to see the uniformity of access to the marked items. Let $T(m)$ be the number of steps taken before a marked item is returned. Conditional on $T(m)$, the access to the marked items is uniform.

We wish to know the statistical properties of Algorithm 1. In particular, we would like to find the mean and variance of number of search steps before an item is returned. Let X_i be the item accessed in the i^{th} step, $i = 1, 2, ...$

2.1 Mean of $T(m)$

Let $T(i)$ be the number of steps taken by the algorithm to return a marked item, assuming the total number of items is i and the items are $1, 2, ..., i$, where $k \leq i \leq m$. Conditional on the first item accessed, denoted by X_1, we have the following iterative relation.

$$\mathbf{E}T(i) = \begin{cases} 1 & \text{if } i = k, \\ \frac{k}{i} + \frac{1}{i}\sum_{j=k+1}^{i}(1 + \mathbf{E}T(j)) & \text{if } k+1 \leq i \leq m. \end{cases} \tag{1}$$

Denote $h(i) = \mathbf{E}T(i)$ for notational simplicity. From (1), we can show

Lemma 2.1 *For $k + 1 \leq i \leq m$,*

$$h(i) = 1 + \frac{1}{k} + \frac{1}{k+1} + ... + \frac{1}{i-1} \tag{2}$$

By approximating the above sum with integral, we get the following bounds for $\mathbf{E}T(m)$ for $1 < k < m$,

$$1 + \ln\frac{m}{k} \leq \mathbf{E}T(m) \leq 1 + \ln\frac{m-1}{k-1} \tag{3}$$

2.2 Variance of $T(m)$

Again by conditioning on the first item accessed, X_1, we have the following iterative relation, for $i = k+1, ..., m$.

$$\mathbf{E}T^2(i) = \frac{k}{i} + \frac{1}{i}\sum_{j=k+1}^{i} \mathbf{E}(1 + T(j))^2 \tag{4}$$

Let us denote $f(i) = \mathbf{E}T^2(i)$. From (4), we have

$$f(i) = \frac{k}{i} + \frac{1}{i}\sum_{j=k+1}^{i} (1 + f(j) + 2h(j)) \tag{5}$$

It is easy to show

Lemma 2.2 *For $k + 1 \le i \le m$,*

$$f(i) = h^2(i) + \frac{1}{k^2} + \frac{1}{(k+1)^2} + ... + \frac{1}{(i-1)^2} + \frac{1}{k} + \frac{1}{k+1} + ... + \frac{1}{i-1} \tag{6}$$

Corollary 2.3 *For $m > k$,*

$$\textit{Variance}(T(m)) = \frac{1}{k^2} + \frac{1}{(k+1)^2} + ... + \frac{1}{(m-1)^2} + \frac{1}{k} + \frac{1}{k+1} + ... + \frac{1}{m-1} \tag{7}$$

For $1 < k < m$, reasonable bounds for Variance$(T(m))$ are

$$\ln\frac{m}{k} + \frac{1}{k} - \frac{1}{m} \le \text{Variance}(T(m)) \le \ln\frac{m-1}{k-1} + \frac{1}{k-1} - \frac{1}{m-1} \tag{8}$$

For large m and $k \ll m$,

$$\text{Variance}(T(m)) \approx \ln\frac{m}{k} \tag{9}$$

We have established that, in the randomized binary search algorithm, the mean of $T(m)$ is roughly $\ln\frac{m}{k}$ and the standard deviation of $T(m)$ is $\sqrt{\ln\frac{m}{k}}$. Note that $\ln\frac{m}{k} = \log_2\frac{m}{k}/\log_2 e = 0.693\log_2\frac{m}{k}$. Hence, it takes fewer steps (in expectation) to find a marked item using the randomized algorithm than using the deterministic binary search algorithm, and the statistical fluctuation of the former is not large.

3 Continuous Case - Algorithm 2

In the previous section, we have derived simple expressions for the first and second order statistics of $T(m)$, the first time a marked item is accessed. We don't have a simple expression for the distribution of $T(m)$. It is not difficult to derive an iterative relationship of $P\{T(i) = l\}$ similar to (1) and (4). The difficulty lies in that there appears to be no easy way to simplify the expansion from the iteration. We will resort to a continuous and scaled version of Algorithm 1 to accomplish this task approximately. In this algorithm, the complete interval is $[0, 1]$ on which the interval $[0, a]$ is marked, where $0 < a \leq 1$ and a is unknown. Otherwise, the algorithm proceeds in the same way as Algorithm 1. Specifically, initialize $x \Leftarrow 1$. In each step, a number y is chosen uniformly at random on $[0, x]$. If y is marked, the algorithm returns y; otherwise, the search continues with $x \Leftarrow y$. This is known as Algorithm 2.

Let the random variable $T(x)$ be the number of steps it takes before the algorithm returns the desired $y \in [0, a]$, given the initial interval is $[0, x]$, where $a \leq x \leq 1$. We wish to compute the distribution of $T(1)$ and its first and second order statistics.

3.1 Distribution of $T(1)$

First, when $x = a$, $P\{T(x) = 1\} = 1$. With the convention $0^0 = 1$, and by conditional on the first jump position, we can get

Lemma 3.1 *For $i \geq 1$ and $a \leq x \leq 1$,*

$$P\{T(x) = i\} = \frac{1}{(i-1)!} \frac{a}{x} (\ln \frac{x}{a})^{i-1} \tag{10}$$

A Poisson random variable Z with mean λ has the distribution $P\{Z = i\} = \frac{e^{-\lambda}\lambda^i}{i!}$, for $i = 0, 1, 2, \ldots$. The variance of Z is also λ. Note that $T(x) - 1$ is a Poisson random variable with mean $\ln \frac{x}{a}$. Hence,

$$\mathbf{E}T(1) = \ln \frac{1}{a} + 1 \tag{11}$$

$$\text{variance}(T(1)) = \ln \frac{1}{a} \tag{12}$$

3.2 Expected Hits to an Arbitrary Item

In the discrete case, we have established that Algorithm 1 accesses and selects one of the k marked items uniformly at random. In other words, each marked items is accessed $1/k$ times on average by the end of the algorithm. However, the access pattern to the un-marked items is not uniform. In the aforementioned load-balancing application on peer-to-peer networks, we wish to load-balance computer 1 to k by choosing one of them uniformly for file downloading. We also wish not to overload other computers corresponding to the un-marked items with excessive query traffic. Therefore, our next question is, by the end of the algorithm, how many times item i has been accessed, where $k < i \leq m$.

We will again work with the continuous case for convenience. Again, X_i is the position of the i^{th} jump in Algorithm 2, $i = 1, 2, \ldots$. To simplify the notation, let $T = T(1)$. Let us consider the stopped process, $X_1, X_2, ..., X_T$. For each $0 \leq y \leq 1$, let $N(y)$ be the number of X_i's less than or equal to y in the stopped process. That is

$$N(y) = |\{i : X_i \leq y, i = 1, 2, ..., T\}| = \sum_{i=1}^{T} 1_{(X_i \leq y)}$$

where the indicator function $1_{(X_i \leq y)}$ is equal to 1 when $X_i \leq y$, and equal to 0 otherwise. Let $n(y) = \frac{d\mathbf{E}N(y)}{dy}$, and call it *hit density*. It is a kind of “density” in the sense that the expected number of hits on a small interval $[y, y + \Delta y]$ is approximately $n(y)\Delta y$. It can be shown that

Theorem 3.2

$$n(y) = \begin{cases} \frac{1}{a} & \text{for } 0 \leq y \leq a \\ \frac{1}{y} & \text{for } a < y \leq 1 \end{cases} \tag{13}$$

Proof: When $0 \leq y \leq a$,

$$\mathbf{E}N(y) = P\{X_T \leq y\} = \frac{y}{a} \tag{14}$$

Hence, $n(y) = \frac{1}{a}$. We will next focus on the case $a < y \leq 1$. In this case,

$$N(y) = \sum_{i=1}^{T-1} 1_{(X_i \leq y)}$$

$$n(y) = \sum_{j=2}^{\infty} P\{T = j\} \sum_{i=1}^{j-1} p(X_i = y|T = j) \tag{15}$$

where $p(X_i = y|T = j)$ denotes the conditional density of X_i given $T = j$. To compute this conditional density, we start with the joint density. For $a < x_{j-1} \leq x_{j-2} \leq ... \leq x_1 \leq 1$,

$$\begin{aligned}
&p(T = j, X_1 = x_1, ..., X_{j-1} = x_{j-1}) \\
=&p(T = j|X_{j-1} = x_{j-1})p(X_{j-1} = x_{j-1}|X_{j-2} = x_{j-2})...p(X_2 = x_2|X_1 = x_1)p(X_1 = x_1) \\
=&\frac{a}{x_{j-1}} \frac{1}{x_{j-2}} ... \frac{1}{x_1}
\end{aligned} \tag{16}$$

Hence,

$$p(X_1 = x_1, ..., X_{j-1} = x_{j-1}|T = j) = \frac{a}{x_{j-1}} \frac{1}{x_{j-2}} ... \frac{1}{x_1} / P\{T = j\} \tag{17}$$

We now compute the marginal density $p(X_i = x_i|T = j)$ for $1 \leq i \leq j - 1$.

$$\begin{aligned}
&p(X_i = x_i|T = j) \\
=&\frac{a}{P\{T = j\}} \int_{a < x_{j-1} \leq ... \leq x_{i+1} \leq x_i \leq x_{i-1} \leq ... \leq x_1 \leq 1} \frac{1}{x_{j-1}} ... \frac{1}{x_{i+1}} \frac{1}{x_i} \frac{1}{x_{i-1}} ... \frac{1}{x_1} dx_{j-1}...dx_{i+1}dx_{i-1}...dx_1 \\
=&\frac{a}{P\{T = j\}} \frac{1}{x_i} \int_{x_i}^{1} \frac{dx_1}{x_1} \int_{x_i}^{x_1} \frac{dx_2}{x_2} ... \int_{x_i}^{x_{i-2}} \frac{dx_{i-1}}{x_{i-1}} \int_{a}^{x_i} \frac{dx_{i+1}}{x_{i+1}} \int_{a}^{x_{i+1}} \frac{dx_{i+2}}{x_{i+2}} ... \int_{a}^{x_{j-2}} \frac{dx_{j-1}}{x_{j-1}}
\end{aligned} \tag{18}$$

By simple induction, it is easy to show

$$\int_{a}^{x_i} \frac{dx_{i+1}}{x_{i+1}} \int_{a}^{x_{i+1}} \frac{dx_{i+2}}{x_{i+2}} ... \int_{a}^{x_{j-2}} \frac{dx_{j-1}}{x_{j-1}} = \frac{1}{(j-1-i)!} (\ln \frac{x_i}{a})^{j-1-i} \tag{19}$$

and

$$\int_{x_i}^{1} \frac{dx_1}{x_1} \int_{x_i}^{x_1} \frac{dx_2}{x_2} ... \int_{x_i}^{x_{i-2}} \frac{dx_{i-1}}{x_{i-1}} = \frac{1}{(i-1)!} (\ln \frac{1}{x_i})^{i-1} \tag{20}$$

Finally, we get

$$p(X_i = x_i|T = j) = \frac{a}{P\{T = j\}} \frac{1}{x_i} \frac{1}{(j-1-i)!} (\ln \frac{x_i}{a})^{j-1-i} \frac{1}{(i-1)!} (\ln \frac{1}{x_i})^{i-1} \tag{21}$$

For $a < y \leq 1$, combining (21) and (15), we have

$$
\begin{aligned}
n(y) &= \sum_{j=2}^{\infty} \frac{a}{y} \sum_{i=1}^{j-1} \frac{1}{(j-1-i)!} (\ln \frac{y}{a})^{j-1-i} \frac{1}{(i-1)!} (\ln \frac{1}{y})^{i-1} \\
&= \frac{a}{y} \sum_{j=2}^{\infty} \frac{1}{(j-2)!} (\ln \frac{y}{a} + \ln \frac{1}{y})^{j-2} \\
&= \frac{a}{y} \exp(\ln \frac{y}{a} + \ln \frac{1}{y}) \\
&= \frac{1}{y}
\end{aligned}
$$

■

From the above lemma, we see that the un-marked region is hit less than the marked region per unit length. Translating this observation to the load-balancing application, we conclude that even though the non-cache servers are not accessed uniformly, each of them is accessed less than any of the cache servers.

Due to the fact that the continuous algorithm approximates the discrete algorithm, Theorem 3.2 should also approximately apply to the discrete case. In figure 1, we plot the simulation results of hit counts to each item for the discrete algorithm, that is, the expected number of hits to each item by the time the algorithm finishes. In the same figure, we also show the function $1/n$, for $1 \leq n \leq m$ and $1/k$. We see that Theorem 3.2 applies very well here.

4 Batch Algorithm A - Algorithm 3

We next consider the following batch version of Algorithm 2, with s as the batch size. The idea is to send s searches together in each step. The first one in the batch that hits the region $[0, a]$ will be returned. Otherwise, another s searches will be sent in the next step. Specifically, initialize $y_0 \Leftarrow 1$. In each search step, generate a batch of s random numbers, $y_1 \in [0, y_0]$, $y_2 \in [0, y_1]$, ..., $y_s \in [0, y_{s-1}]$, each chosen uniformly at random within the respective interval. If at least one of the numbers in the batch lands on $[0, a]$ and y_i, $1 \leq i \leq s$, is the first of these, then return y_i. Otherwise, continue with $y_0 \Leftarrow y_s$. This algorithm will be known as batch algorithm A or Algorithm 3.

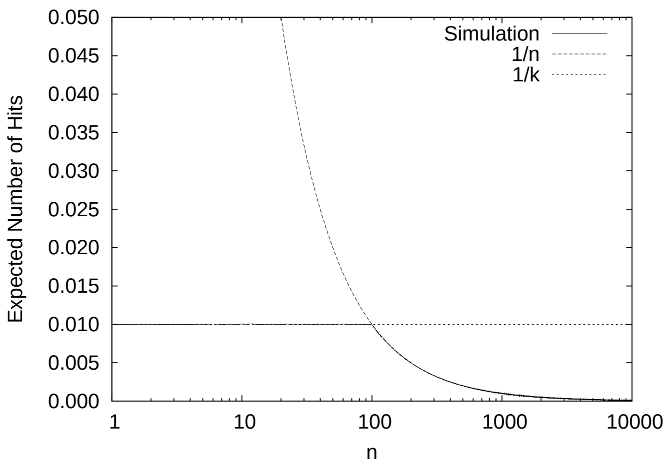

Figure 1: Expected number of hits to each item, for $m = 10000$ and $k = 100$.

Clearly, the number returned by the algorithm is chosen uniformly at random from $[0, a]$, but the number of search steps is cut down by a factor of s. The access pattern to the un-marked region $(a, 1]$ remains the same as in Algorithm 2. However, the access pattern to the marked region $[0, a]$ is no longer uniform. We will proceed to analyze the access pattern to the marked region by computing the hit density $n(y)$.

Theorem 4.1 *For* $0 < y \leq a$,

$$n(y) = \sum_{j=1}^{s} \sum_{i=0}^{s-j} \frac{1}{a} \frac{1}{i!} (\ln \frac{a}{y})^i P\{J = j\} \tag{22}$$

Proof: Suppose the first hit to the marked region $[0, a]$ is the J^{th} number drawn in the batch, where J is a random variable taking values in $\{1, 2, ..., s\}$. That is, y_J is the first number landing in $[0, a]$ in the final batch before Algorithm 3 returns. Then, by (10),

$$\begin{aligned} P\{J = j\} &= P\{T \equiv j \bmod s\} \\ &= \sum_{\{r=si+j:i=0,1,...\}} \frac{a}{(r-1)!} (\ln \frac{1}{a})^{r-1} \\ &= \sum_{i=0}^{\infty} \frac{a}{(si+j-1)!} (\ln \frac{1}{a})^{si+j-1} \end{aligned} \tag{23}$$

As before, we denote $T = T(1)$, the first time Algorithm 2 (not Algorithm 3) enters $[0, a]$. The infinite sum in (23) does not necessarily pose a computational problem because the factorial increases very fast. Typically, the first twenty terms in (23) give enough accuracy for a wide range of values for a.

Now, once y_J hits $[0, a]$, $y_{J+1}, ..., y_s$ will also hit $[0, a]$. Therefore, for $0 < y \le a$,

$$\begin{aligned} n(y) &= \mathbf{E} \sum_{i=0}^{s-J} p(X_{T+i} = y) \qquad (24) \\ &= \sum_{j=1}^{s} \sum_{i=0}^{s-j} p(X_{T+i} = y | J = j) P\{J = j\} \\ &= \sum_{j=1}^{s} \sum_{i=0}^{s-j} p(X_{T+i} = y) P\{J = j\} \qquad (25) \end{aligned}$$

where the expectation is taken over the probability distribution of J, X_i's are as defined before, and $p()$ is the probability density function. We drop the conditioning in (25) because X_T has uniform distribution on $[0, a]$ regardless the value of J, and hence, is independent of J. For $i \ge 1$, X_{T+i} is also independent of J since

$$\begin{aligned} p(X_{T+i} = y | J = j) &= \int p(X_{T+i} = y | X_T = z, J = j) p(X_T = z | J = j) dz \\ &= \int p(X_{T+i} = y | X_T = z) p(X_T = z) dz \end{aligned}$$

From (25), the key is to compute $\sum_{i=0}^{l} p(X_{T+i} = y)$, for $l = 0, 1, ..., s-1$. We claim,

Lemma 4.2 *For $0 < y \le a$ and for $i = 0, 1, ..., s-1$,*

$$p(X_{T+i} = y) = \frac{1}{a} \frac{1}{i!} (\ln \frac{a}{y})^i \qquad (26)$$

Proof: We have already established $p(X_T = y) = \frac{1}{a}$. Let us suppose (26) is true for some $i \ge 0$ and we will show it is true for $i + 1$.

$$\begin{aligned} p(X_{T+i+1} = y) = & \int_y^a p(X_{T+i+1} = y | X_{T+i} = x) p(X_{T+i} = x) dx \\ = & \int_y^a \frac{1}{x} \frac{1}{a} \frac{1}{i!} (\ln \frac{a}{x})^i dx \\ = & \int_{\frac{y}{a}}^1 \frac{1}{a} \frac{1}{z} \frac{1}{i!} (\ln \frac{1}{z})^i dz \\ = & \frac{1}{a} \int_{\frac{y}{a}}^1 (-1)^i \frac{1}{i!} (\ln z)^i d(\ln z) \\ = & \frac{1}{a} (-1)^i \frac{1}{(i+1)!} (\ln z)^{i+1} \Big|_{\frac{y}{a}}^1 \\ = & \frac{1}{a} \frac{1}{(i+1)!} (\ln \frac{a}{y})^{i+1} \end{aligned} \tag{27}$$

In (27), we use the change of variable $x = az$. ■

By Lemma 4.2 and (25), we get

$$n(y) = \sum_{j=1}^{s} \sum_{i=0}^{s-j} \frac{1}{a} \frac{1}{i!} (\ln \frac{a}{y})^i P\{J = j\} \tag{28}$$

■

Note that, for each fi xed j,

$$\lim_{s \to \infty} \sum_{i=0}^{s-j} \frac{1}{a} \frac{1}{i!} (\ln \frac{a}{y})^i = \frac{1}{a} \exp(\ln \frac{a}{y}) = \frac{1}{y}$$

Hence, $\lim_{s \to \infty} n(y) = \frac{1}{y}$. With (23) and (22), we can compute $n(y)$ numerically. Figure 2 shows the numerical results for the case of $a = 0.1$, and for batch sizes $s = 1, 2, 4$ or 8. It also shows the limiting function $1/y$ as the batch size approaches to infi nity. We see that the hit density approaches the limit very fast as the batch size increases. In the limit, the hit densities at small values (e.g., $y = 0.0001$) can be orders of magnitude larger than the case with $s = 1$.

5 Batch Algorithm B - Algorithm 4

Algorithm 3 with batch size s cuts down the search steps by s, but can increase the hit counts to the region near 0 dramatically. We next consider a second batch algorithm, known as Algorithm 4. The idea

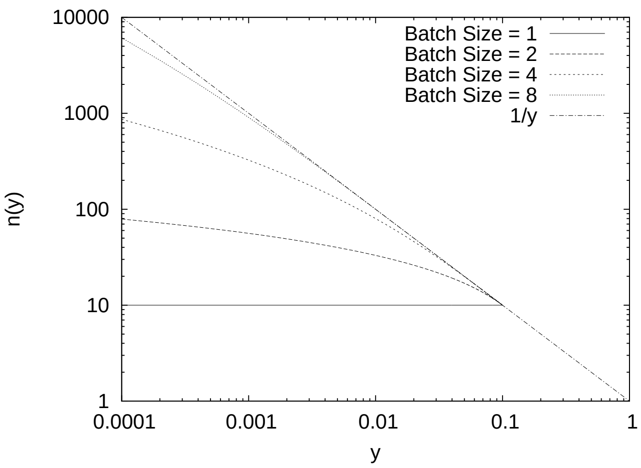

Figure 2: Hit densities different batch sizes. $a = 0.1$.

is to send s searches together in each step, the target of which are all uniformly drawn from the same (current) interval. If any of the searches hits the region $[0, a]$, one of the hits will be randomly picked as the result of the algorithm. Specifically, initialize $x \Leftarrow 1$. In each search step, generate a batch of s random numbers, y_1, y_2, ..., $y_s \in [0, x]$, each chosen uniformly at random on $[0, x]$. If at least one of the numbers in the batch lands on the marked region $[0, a]$, choose one among all y_i's landing on $[0, a]$ uniformly at random and return it. Otherwise, set x to be the minimum of $y_1, \ldots, y_s$ and continue. The algorithm has the properties that, first, the number returned by the algorithm is chosen uniformly at random from $[0, a]$, second, the access pattern to the region $[0, a]$ is uniform, and third, the access pattern to the un-marked region $(a, 1]$ remains the same as in Algorithm 2 except that the magnitude increases by a factor of s. We wish to find the expected number of search steps, again denoted by $\mathbf{E}T$, when the algorithm returns and to compare it with Algorithm 2.

First, we will argue informally that the expected number of search steps, $\mathbf{E}T$, is cut down by a factor of $\ln s$, as compared with Algorithm 2. Then, we will find the exact $\mathbf{E}T$ for several s and its asymptotic value for general s as $1/a$ approaches infinity. In Section 5.4, we will use similar procedures to find the asymptotic variance.

Consider a generic step, where $y_1, y_2, ..., y_s$ are drawn uniformly at random on $[0, x]$. Let $W =$

$\min\{y_1, y_2, ..., y_s\}$. Then,

$$P\{W > y\} = P\{y_1 > y, y_2 > y, ..., y_s > y\} = (1 - \frac{y}{x})^s \tag{29}$$

$$\mathbf{E}W = \int_0^x (1 - \frac{y}{x})^s dy = \frac{x}{s+1} \tag{30}$$

Hence, on average, the minimum of all searches in each step lands at $\frac{x}{s+1}$, where the interval is $[0, x]$. Hence, it takes about $\log_{s+1} \frac{1}{a} = \frac{\ln(1/a)}{\ln(s+1)}$ steps to hit the region $[0, a]$, starting at 1.

5.1 A Martingale Argument for an Upper Bound on $\mathbf{E}T(1)$

Consider the following stochastic process, $\{X_n\}_{n=1}^{\infty}$. In each step $n \geq 1$, $Y_n^{(1)}, Y_n^{(2)}, ..., Y_n^{(s)}$ are chosen uniformly at random on $[0, X_{n-1}]$. For convenience, we let $X_0 = 1$. Let $X_n = \min\{Y_n^{(1)}, Y_n^{(2)}, ..., Y_n^{(s)}\}$. Under this setup, $\{(s+1)^n X_n; n \geq 1\}$ is a Martingale. To see this, for $n \geq 1$,

$$\begin{aligned}
&\mathbf{E}[(s+1)^{n+1} X_{n+1} \mid (s+1)^n X_n, ..., (s+1) X_1] \\
=&\mathbf{E}[(s+1)^{n+1} X_{n+1} \mid (s+1)^n X_n] \\
=&(s+1)^{n+1} \frac{X_n}{s+1} \\
=&(s+1)^n X_n
\end{aligned} \tag{31}$$

The step to obtain (31) uses the result in (30). Let T be the first time $\{X_n; n \geq 1\}$ enters $[0, a]$, which is a stopping time. By the Martingale stopping theorem [5],

$$\mathbf{E}[(s+1)^T X_T] = \mathbf{E}[(s+1) X_1] = 1 \tag{32}$$

Now,

$$\begin{aligned}
\mathbf{E}[(s+1)^T X_T] &= \mathbf{E}[\mathbf{E}[(s+1)^T X_T \mid T]] \\
&= \mathbf{E}[(s+1)^T \mathbf{E}[X_T \mid T]]
\end{aligned} \tag{33}$$

But,

$$\mathbf{E}[X_T \mid T] \geq \frac{a}{s+1} \tag{34}$$

because for any fixed $n \geq 2$ and $a \leq x \leq 1$,

$$\mathbf{E}[X_T \mid T = n, X_{n-1} = x] = \frac{x}{s+1} \geq \frac{a}{s+1}$$

Combining (32), (33) and (34), we get,

$$\frac{a}{s+1}\mathbf{E}[(s+1)^T] \leq 1 \tag{35}$$

By Jensen's inequality, we have

$$\frac{a}{s+1}(s+1)^{\mathbf{E}T} \leq \frac{a}{s+1}\mathbf{E}[(s+1)^T] \leq 1 \tag{36}$$

Hence,

$$\mathbf{E}T \leq \log_{s+1}\frac{s+1}{a} = 1 + \frac{\ln\frac{1}{a}}{\ln(s+1)} \tag{37}$$

Compared with the result for $s = 1$, as shown in (11), the expected number of searches has been cut down by at least a factor $\ln(s+1)$. We will show that the upper bound is essentially tight in the asymptotic regime.

5.2 Exact Computation of $\mathbf{E}T$

For ease of presentation, let us normalize the marked region to be $[0, 1]$ and let us begin the batch algorithm at $x > 1$. Again, let $h(x) = \mathbf{E}T(x)$, where $T(x)$ is the number of search steps before hitting the marked region, given that the initial search interval is $[0, x]$. From (29), in the first step, $X_1 = \min\{Y_1^{(1)}, Y_1^{(2)}, ..., Y_1^{(s)}\}$ has density

$$f_{X_1}(y) = s(1 - \frac{y}{x})^{s-1}\frac{1}{x} \tag{38}$$

where $s \geq 1$ and $0 \leq y \leq x$. Conditional on X_1, we get

$$h(x) = 1 + \int_1^x \frac{s}{x}(1 - \frac{y}{x})^{s-1}h(y)dy \tag{39}$$

Here, we assume $h(1) = 1$. The integral equation (39) belongs to the family of Volterra equations of the second kind. Several general techniques exist for solving it, but typically involves repeated integration and summing of an infinite series [3]. In practice, the procedures quickly become too complicated to be carried out by hand, and any partial result provides no apparent insight in how the final result looks like. We will take a different approach by converting the integral equation into a differential equation. This is possible because of the particular form of equation (39), which we rewrite as follows.

$$x^s h(x) = x^s + \int_1^x s(x-y)^{s-1} h(y) dy \tag{40}$$

Take derivative with respect to x s times, we get

$$\frac{d^s(x^s h(x))}{dx^s} = s! + s! h(x) \tag{41}$$

After expanding the left hand side, we get

$$\sum_{i=0}^{s} \binom{s}{i} \frac{s!}{i!} x^i h^{(i)}(x) = s! + s! h(x) \tag{42}$$

where $h^{(i)}(x)$ is the i^{th} derivative of h. The above equation further simplifies to

$$\sum_{i=1}^{s} \binom{s}{i} \frac{s!}{i!} x^i h^{(i)}(x) = s! \tag{43}$$

The initial conditions can be derived from (39) and its derivatives of orders up to $s-1$, evaluated at $x=1$. By assumption, $h(1)=1$. The first derivative can be written as

$$\begin{aligned} h'(x) &= \int_1^x \frac{d}{dx}(\frac{s}{x}(1-\frac{y}{x})^{s-1}) h(y) dy + \frac{s}{x}(1-\frac{x}{x})^{s-1} \\ &= \int_1^x \frac{d}{dx}(\frac{s}{x}(1-\frac{y}{x})^{s-1}) h(y) dy \end{aligned} \tag{44}$$

When evaluated at $x=1$, $h'(1)=0$. It is easy to see that, for $1 \le k \le s-1$,

$$h^{(k)}(x) = \int_1^x \frac{d^k}{dx^k}(\frac{s}{x}(1-\frac{y}{x})^{s-1}) h(y) dy \tag{45}$$

Hence, the initial conditions are

$$h(1) = 1 \tag{46}$$

$$h^{(k)}(1) = 0 \quad \text{for } 1 \le k \le s-1 \tag{47}$$

We now investigate the solutions for the differential equation (43) with the initial conditions (46) and (47). For each fixed s, we will see later that this amounts to solving a polynomial equation of degree s with constant coefficients. For several small s, one can find exact solutions. The process of finding exact solutions is fundamentally limited by whether one can find the roots of the underlying polynomial of degree s algebraically with finite additions, multiplication, division and taking root. It is known that, in general, this can be done only for polynomials of degree 4 or less. In our case, due to the special structure of the polynomial, we can do so for $s \leq 7$. For larger s, one must find the roots of the corresponding polynomial numerically. In the following result (48), we give the solutions for $s = 2, 3$ and 4, which can be obtained by either hand analysis using the procedure introduced later in Section 5.3, or by using mathematical software such as Mathematica[2]. Note that, as x becomes large, $h(x)$ approaches $\frac{2}{3}\ln x + \frac{7}{9}$, $\frac{6}{11}\ln x + \frac{85}{121}$, and $\frac{12}{25}\ln x + \frac{83}{125}$ for $s = 2, 3$ and 4, respectively. Figure 3 shows $h(x)$ graphically.

$$h(x) = \begin{cases} \frac{2}{3}\ln x + \frac{7}{9} + \frac{2}{9x^3} & s = 2 \\ \frac{6}{11}\ln x + \frac{85}{121} + \frac{36}{121x^3}\cos(\sqrt{2}\ln x) & \\ \quad + \frac{21\sqrt{2}}{121x^3}\sin(\sqrt{2}\ln x) & s = 3 \\ \frac{12}{25}\ln x + \frac{83}{125} + \frac{6}{25x^{5/2}}\cos(\frac{\sqrt{15}}{2}\ln x) & \\ \quad + \frac{2\sqrt{15}}{25x^{5/2}}\sin(\frac{\sqrt{15}}{2}\ln x) + \frac{12}{125x^5} & s = 4 \end{cases} \tag{48}$$

5.3 Asymptotic Behavior of $\mathrm{E}T(x)$

For general s, since it is difficult to find the exact result for $h(x)$, we will try to find the asymptotic result for large x. In particular, we wish to find the constant before $\ln x$ in the expressions such as (48).

Equation (43) is a linear differential equation. We will need to first find a particular solution, and

[2] Software from Wolfram Research

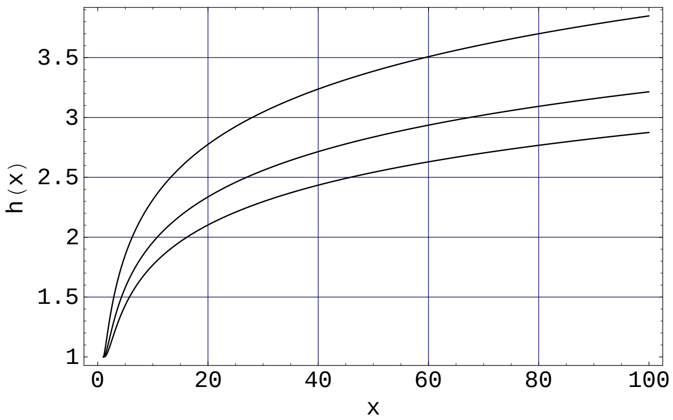

Figure 3: $h(x)$. The bottom, middle and top curves are for $s = 2, 3$ and 4, respectively.

then, the independent solutions for the associated homogeneous equation

$$\sum_{i=1}^{s} \binom{s}{i} \frac{s!}{i!} x^i h^{(i)}(x) = 0 \tag{49}$$

The complete solution is the sum of the particular solution and the linear combination of all independent homogeneous solutions, with the latter giving a general solution to the homogeneous equation. We will show that, as x becomes large, any homogeneous solution tends to a constant, and that, $h(x)$ is essentially the particular solution plus that constant. More formally, we will prove the following main theorem.

Theorem 5.1 *As $x \to \infty$, $h(x) - \eta \ln x$ tends to a finite constant, where $\eta = 1/\sum_{i=1}^{s} \frac{1}{i}$.*

An alternative statement is that, as $x \to \infty$, $h(x) = \eta \ln x + C + o(1)$ for some fixed constant C. Note that $\ln(s+1) \leq \sum_{i=1}^{s} \frac{1}{i} \leq 1 + \ln s$.

To prove Theorem 5.1, we will first show that $\eta \ln x$ is a particular solution to (43), and then, we will prove that any linear combination of the independent homogeneous solutions tends to a finite constant as x approaches infinity.

5.3.1 Particular Solution

We only need to guess a particular solution. We claim that a particular solution has the form

$$h_{part}(x) = \eta \ln x \tag{50}$$

By substituting $h_{part}(x)$ and its derivatives into (43), we can determine the coefficient η.

$$\eta = \frac{1}{\sum_{i=1}^{s} \binom{s}{i} \frac{1}{i}(-1)^{i-1}} \tag{51}$$

Interestingly, the following equality holds. (See [2], page 5.)

$$\sum_{i=1}^{s} \binom{s}{i} \frac{1}{i}(-1)^{i-1} = \sum_{i=1}^{s} \frac{1}{i} \tag{52}$$

Hence,

$$\eta = \frac{1}{\sum_{i=1}^{s} \frac{1}{i}} \tag{53}$$

5.3.2 Homogeneous Solutions

The homogeneous equation (49) is an Euler equation [4], which has the following general form

$$a_n x^n h^{(n)} + a_{n-1} x^{n-1} h^{(n-1)} + ... + a_1 x h' + a_0 h = 0 \tag{54}$$

With the substitution $x = e^t$, an Euler equation can be transformed into a linear differential equation with constant coefficients, which can then be solved if the roots of the corresponding characteristic equation can be found. As an example, consider the homogeneous equation (49) for the case $s = 2$.

$$x^2 h''(x) + 4xh'(x) = 0 \tag{55}$$

Given $x = e^t$, let $\phi(t) = h(e^t)$. It is easy to derive the following expressions.

$$h'(x) = e^{-t}\phi'(t) \tag{56}$$

$$h''(x) = e^{-2t}(\phi''(t) - \phi'(t)) \tag{57}$$

Substituting $x = e^t$, (56) and (57) into (55), we get

$$\phi''(t) + 3\phi'(t) = 0 \tag{58}$$

The solutions are $C_1 e^{-3t} + C_2$ for some constants C_1 and C_2. Hence, the homogeneous solutions to (55) are

$$h_{hom} = C_1 x^{-3} + C_2 \tag{59}$$

for some constants C_1 and C_2.

It is known that after the substitution $x = e^t$, the Euler equation (54) becomes the following linear differential equation with constant coeffi cients [4].

$$\sum_{i=1}^{n} a_i D(D-1)...(D-i+1)\phi = -a_0\phi \tag{60}$$

where D is the derivative operator $\frac{d}{dx}$. In our case, corresponding to (49), we get the following equation

$$\sum_{i=1}^{s} \binom{s}{i} \frac{s!}{i!} D(D-1)...(D-i+1)\phi = 0 \tag{61}$$

The corresponding characteristic equation is

$$\sum_{i=1}^{s} \binom{s}{i} \frac{s!}{i!} \lambda(\lambda-1)...(\lambda-i+1) = 0 \tag{62}$$

Lemma 5.2

$$\sum_{i=1}^{s} \binom{s}{i} \frac{s!}{i!} \lambda(\lambda-1)...(\lambda-i+1) = (\lambda+1)(\lambda+2)...(\lambda+s) - s! \tag{63}$$

Proof:

$$\begin{aligned}
\sum_{i=1}^{s}\binom{s}{i}\frac{s!}{i!}\lambda(\lambda-1)...(\lambda-i+1) =& s!\sum_{i=1}^{s}\binom{s}{i}\binom{\lambda}{i} \\
=& s!\sum_{i=1}^{s}\binom{s}{s-i}\binom{\lambda}{i} \\
=& s!(\sum_{i=0}^{s}\binom{s}{s-i}\binom{\lambda}{i}-1) \\
=& s!\binom{s+\lambda}{s}-s! \qquad (64) \\
=& (\lambda+1)(\lambda+2)...(\lambda+s)-s!
\end{aligned}$$

In (64), we have used the known identity [8],

$$\binom{y+z}{n}=\sum_{j=0}^{n}\binom{y}{n-j}\binom{z}{j} \qquad (65)$$

■

Let us define

$$g(\lambda)=(\lambda+1)(\lambda+2)...(\lambda+s)-s! \qquad (66)$$

The characteristic equation is

$$g(\lambda)=0 \qquad (67)$$

We wish to characterize the roots (also known as zeros) of the polynomial $g(\lambda)$.

Theorem 5.3 *Except the root $\lambda=0$, all other roots of the polynomial $g(\lambda)$ have negative real parts.*

To prove this, we need a theorem known as the Zero Exclusion Principle [1]. Let us consider a family of parameterized polynomials. Let $\delta(x,p)$ be a polynomial whose coefficients depend continuously on the parameter p, for $p\in\Omega\subset\mathbb{R}^l$, where l is a natural number. That is, $\delta(x,p)$ can be written as

$$\delta(x,p)=\sum_{i=0}^{n}a_i(p)x^i$$

where each $a_i(p)$ is a continuous function of p on Ω.

Let us denote the family of polynomials

$$\Delta(x) = \{\delta(x, p) \mid p \in \Omega\} \tag{68}$$

We say the family of polynomials in (68) is of *constant degree* if $a_n(p) \neq 0$ for any $p \in \Omega$. Suppose we are given a region of stability, denoted by S, which is an open subset of the complex plane $\mathbb{C}$. A polynomial is called a *stable polynomial* if its roots all fall into the stability region S. For our purpose, S is the open left half plane, i.e., $\{z \in \mathbb{C} \mid \Re(z) < 0\}$, where $\Re$ stands for the real part. Let us denote the boundary of S by ∂S. More formally, Let U^o be the interior of the closed set $\mathbb{C}\backslash S$. Then,

$$S \cup \partial S \cup U^o = \mathbb{C} \qquad S \cap \partial S = \partial S \cap U^o = \emptyset$$

Theorem 5.4 (Zero Exclusion Principle [1]) *Assume that the family of polynomials in (68) is of constant degree, contains at least one stable polynomial, and Ω is piecewise connected. Then the entire family is stable if and only if*

$$0 \notin \Delta(x^*), \quad \textit{for all } x^* \in \partial S$$

Returning to our problem, ∂S is the imaginary axis. Let

$$\delta(\lambda, p) = (\lambda + 1)(\lambda + 2)...(\lambda + s) - p$$

Note that $g(\lambda) = \delta(\lambda, s!)$. Let $\Omega = [0, s!)$, and let $\Delta(\lambda) = \{\delta(\lambda, p) \mid p \in [0, s!)\}$. We will show that

Lemma 5.5 *The family of polynomials $\Delta(\lambda)$ is stable.*

Proof: First,

$$\delta(\lambda, 0) = (\lambda + 1)(\lambda + 2)...(\lambda + s)$$

is clearly a stable polynomial. Next, let $\lambda^* \in \partial S$. Write $\lambda^* = iw$, for $w \in \mathbb{R}$ and $i = \sqrt{-1}$. Then,

$$\begin{aligned} \delta(iw, p) &= (iw + 1)(iw + 2)...(iw + s) - p \\ &= \sqrt{w^2 + 1}\sqrt{w^2 + 2^2}...\sqrt{w^2 + s^2}e^{i\theta(w)} - p \end{aligned}$$

where

$$\theta(w) = tan^{-1}(w) + tan^{-1}(w/2) + ... + tan^{-1}(w/s)$$

clearly, for $p \in [0, s!)$,

$$|\sqrt{w^2+1}\sqrt{w^2+2^2}...\sqrt{w^2+s^2}e^{i\theta(w)}| > p$$

Hence, $\delta(iw, p) \neq 0$ for $p \in [0, s!)$ and for all real w. We can apply the Zero Exclusion Principle and conclude that the family of polynomials $\Delta(\lambda)$ is stable. ■

Note that $g(\lambda)$ is not a stable polynomial, since $\lambda = 0$ is a root. We will next prove Theorem 5.3.

Proof: [of Theorem 5.3] We first show 0 is the only root of $g(\lambda)$ on the imaginary axis.

$$\begin{aligned} g(iw) &= (iw+1)(iw+2)...(iw+s) - s! \\ &= \sqrt{w^2+1}\sqrt{w^2+2^2}...\sqrt{w^2+s^2}e^{i\theta(w)} - s! \end{aligned}$$

Clearly $g(iw) = 0$ if and only if $w = 0$.

Next, suppose $g(\lambda)$ has a root, say z, with real part greater than 0. Let $r = \Re(z)$. Consider a small $\epsilon > 0$ and the polynomial $\delta(\lambda, s! - \epsilon)$. Let $\lambda_i(\epsilon)$ be the roots of $\delta(\lambda, s! - \epsilon)$, for $i = 1, 2, ..., s$. At this point, we do not care whether some of the roots are multiple roots. By Lemma 5.5, $\Re(\lambda_i(\epsilon)) < 0$ for all i. Hence, $\Re(z) - \Re(\lambda_i(\epsilon)) > r$. This is true regardless how small the positive ϵ is. This violates the continuity of roots of polynomials under continuous change of the coefficients. (See [1], Theorem 1.3 on page 32.) ■

Theorem 5.6 *As $x \to \infty$, any solution of the homogeneous equation (49) converges to a finite constant.*

Proof: After the substitution $x = e^t$, equation (49) becomes the homogeneous linear differential equation (61) with constant coefficients. By the theory on such differential equations [6], the general solution to (61) is the linear combination of the s independent solutions corresponding to the roots of the characteristic equation (67). Each real root λ with multiplicity m gives m independent solutions

$$e^{\lambda t}, te^{\lambda t}, ..., t^{m-1}e^{\lambda t} \tag{69}$$

Each pair of complex conjugate roots $\alpha + i\beta$ and $\alpha - i\beta$, each with multiplicity m, give $2m$ independent solutions.

$$\begin{aligned} &e^{\alpha t}\cos(\beta t), te^{\alpha t}\cos(\beta t), ..., t^{m-1}e^{\alpha t}\cos(\beta t) \\ &e^{\alpha t}\sin(\beta t), te^{\alpha t}\sin(\beta t), ..., t^{m-1}e^{\alpha t}\sin(\beta t) \end{aligned} \tag{70}$$

After converting the variable t back to x by $t = \ln x$, the solutions in (69) become

$$x^{\lambda}, x^{\lambda}\ln x, ..., x^{\lambda}(\ln x)^{m-1} \tag{71}$$

The solutions in (70) become

$$\begin{aligned} &x^{\alpha}\cos(\beta \ln x), (\ln x)x^{\alpha}\cos(\beta \ln x), ..., (\ln x)^{m-1}x^{\alpha}\cos(\beta \ln x) \\ &x^{\alpha}\sin(\beta \ln x), (\ln x)x^{\alpha}\sin(\beta \ln x), ..., (\ln x)^{m-1}x^{\alpha}\sin(\beta \ln x) \end{aligned} \tag{72}$$

The general solution to the homogeneous equation (49) is the linear combination of independent solutions in (71) and (72)

In our case, 0 is a simple real root of the characteristic equation (67) because we can factor out λ from the left hand side of (67), $g(\lambda)$, but not λ^2. Hence, the constant function 1 is an independent solution. By Theorem 5.3, all other roots have negative real parts. Hence, as $x \to \infty$, the corresponding independent solutions all tend to zero. ■

We now prove the main theorem of Section 5.3

Proof: [of Theorem 5.1] The complete solution to (43) is a linear combination of the independent homogeneous solutions plus a particular solution. By (50) and (53), $\eta \ln x$ is a particular solution where $\eta = 1/\sum_{i=1}^{s}\frac{1}{i}$. By Theorem 5.6, any such complete solution minus $\eta \ln x$ tends to a finite constant. ■

We next give a refinement that is relevant to the speed of convergence for Theorem 5.6. We state without a proof the following theorem.

Theorem 5.7 *All roots of the characteristic equation (67) are simple. When s is odd, the solution to the* *homogeneous equation (49) has the form*

$$h_{homo}(x) = C_1 + \sum_{j=1}^{(s-1)/2} D_{j,1} x^{\alpha_j} \cos(\ln(\beta_j x)) + \sum_{j=1}^{(s-1)/2} D_{j,2} x^{\alpha_j} \sin(\ln(\beta_j x)) \tag{73}$$

When s is even, the solution has the form

$$h_{homo}(x) = C_1 + C_2 x^{-s-1} + \sum_{j=1}^{s/2-1} D_{j,1} x^{\alpha_j} \cos(\ln(\beta_j x)) + \sum_{j=1}^{s/2-1} D_{j,2} x^{\alpha_j} \sin(\ln(\beta_j x)) \tag{74}$$

In (73) and (74), for all j, $\alpha_j \pm i\beta_j$ are the distinct complex roots of the characteristic equation (67). Also, for all j, $\alpha_j < 0$ and $\beta_j \neq 0$; C_1, C_2, $D_{j,1}$ and $D_{j,2}$ are constants.

5.4 Asymptotic Variance of $T(x)$

Similar analysis can be carried out for the variance of $T(x)$. The main result of this section is the following theorem.

Theorem 5.8 *As $x \to \infty$, the variance of $T(x)$ satisfies*

$$\textit{Variance}(T(x)) = L(s) \ln x + C + o(1) \tag{75}$$

where

$$L(s) = \frac{\sum_{i=1}^{s} \frac{1}{i^2}}{(\sum_{i=1}^{s} \frac{1}{i})^3} \tag{76}$$

and C is a fixed constant.

Proof: Let us consider the second moment of $T(x)$, $\mathbf{E}T^2(x)$. Conditional on X_1, the location chosen by the first step of the algorithm, we have

$$\mathbf{E}T^2(x) = 1 \cdot P\{T(x) = 1\} + \int_1^x f_{X_1}(y) \mathbf{E}(T(y)+1)^2 dy$$

where $f_{X_1}(y)$ is the probability density function of X_1, given by (38). Again denote $\mathbf{E}T^2(x)$ by $f(x)$. We can rewrite the above as

$$
\begin{aligned}
f(x) &= P\{T(x)=1\} + \int_1^x f_{X_1}(y)(f(y)+2h(y)+1)dy \\
&= 1 + \int_1^x f_{X_1}(y)(f(y)+2h(y))dy \\
&= 1 + \int_1^x \frac{s}{x}(1-\frac{y}{x})^{s-1} f(y)dy + \int_1^x \frac{s}{x}(1-\frac{y}{x})^{s-1} 2h(y)dy \\
&= 1 + \int_1^x \frac{s}{x}(1-\frac{y}{x})^{s-1} f(y)dy + 2h(y) - 2 \qquad (77)
\end{aligned}
$$

where in the last step, we have used (39). We can rewrite (77) as,

$$x^s f(x) = \int_1^x s(x-y)^{s-1} f(y)dy + 2x^s h(y) - x^s \qquad (78)$$

Taking derivative s times on both sides of (78) and use (41), we get

$$\sum_{i=0}^{s} \binom{s}{i} \frac{s!}{i!} x^i f^{(i)}(x) = s!f(x) + s! + 2s!h(x) \qquad (79)$$

Equation (79) can be simplified as

$$\sum_{i=1}^{s} \binom{s}{i} \frac{s!}{i!} x^i f^{(i)}(x) = s! + 2s!h(x) \qquad (80)$$

The initial conditions can be obtained from the integral equation (77) and by repeatedly taking its derivatives up to $s-1$ times.

$$f(1) = 1 \qquad (81)$$

$$f^{(k)}(1) = 0 \quad \text{for } 1 \le k \le s-1 \qquad (82)$$

Note that (80) and (43) are similar linear differential equations, with identical left hand side. The corresponding homogenous equations, and therefore, the general solutions to the homogeneous equations are identical, given by the expressions in Theorem 5.7 (with h replaced by f for the case of (80)). In the following, we will outline a procedure for finding a particular solution to (80), known as the *method of undetermined coefficients* [6].

We will work with the version of (80) after the substitution $x = e^t$. Denote $\phi_f(t) = f(e^t)$ and $\phi_h(t) = h(e^t)$. Then, (80) becomes the following linear differential equation with constant coefficients.

$$g(D)\phi_f(t) = s! + 2s!\phi_h(t) \tag{83}$$

where the polynomial $g(\lambda)$ is given by (66), and $D = \frac{d}{dx}$.

As before, let us write the roots of the polynomial $g(\lambda)$ as, for odd s,

$$0,\ \alpha_j + i\beta_j \tag{84}$$

where $j = 1, 2, ..., (s-1)/2$; and for even s,

$$0,\ -s-1,\ \alpha_j + i\beta_j \tag{85}$$

where $j = 1, 2, ..., s/2 - 1$. Recall that these roots are all distinct, simple, $\alpha_j < 0$ and $\beta_j \neq 0$ for all j. By combining the homogeneous solution (73) or (74) with the particular solution (50), and replacing x with e^t, we get the function $\phi_h(t)$. For odd s,

$$\phi_h(t) = \eta t + A_1 + \sum_{j=1}^{(s-1)/2} B_{j,1} e^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{(s-1)/2} B_{j,2} e^{\alpha_j t}\sin(\beta_j t) \tag{86}$$

and for even s,

$$\phi_h(t) = \eta t + A_1 + A_2 e^{(-s-1)t} + \sum_{j=1}^{s/2-1} B_{j,1} e^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{s/2-1} B_{j,2} e^{\alpha_j t}\sin(\beta_j t) \tag{87}$$

Here, the constants $A_1, A_2, B_{j,1}$ and $B_{j,2}$, for all j, are fixed constants chosen to satisfy the initial conditions (46) and (47). Note that $\phi_h(t)$ satisfies

$$g(D)\phi_h(t) = s! \tag{88}$$

Hence, it satisfies

$$Dg(D)\phi_h(t) = 0 \tag{89}$$

The right hand side of (83) satisfies

$$Dg(D)(s! + 2s!\phi_h(t)) = 0 \tag{90}$$

Hence, any particular solution of (83) must satisfy

$$Dg^2(D)\phi_f(t) = 0 \tag{91}$$

Consider the characteristic equation for (91),

$$\lambda g^2(\lambda) = 0 \tag{92}$$

The roots of $\lambda g^2(\lambda)$ are the same as the roots of $g(\lambda)$, except that 0 has multiplicity 3 and all other roots have multiplicity 2. Hence, the independent solutions for (91) are, for odd s,

$$\begin{aligned}&1,\ t,\ t^2,\ e^{\alpha_j t}\cos(\beta_j t),\ e^{\alpha_j t}\sin(\beta_j t)\\ &te^{\alpha_j t}\cos(\beta_j t),\ te^{\alpha_j t}\sin(\beta_j t)\end{aligned} \tag{93}$$

where $j = 1, 2, ..., (s-1)/2$, and for even s,

$$\begin{aligned}&1,\ t,\ t^2,\ e^{(-s-1)t},\ te^{(-s-1)t},\\ &e^{\alpha_j t}\cos(\beta_j t),\ e^{\alpha_j t}\sin(\beta_j t)\\ &te^{\alpha_j t}\cos(\beta_j t),\ te^{\alpha_j t}\sin(\beta_j t)\end{aligned} \tag{94}$$

where $j = 1, 2, ..., s/2 - 1$.

Hence, any solution to (83) can be written as, for odd s,

$$\begin{aligned}\phi_f(t) = C_1 + C_2 t + C_3 t^2 + \sum_{j=1}^{(s-1)/2} D_{j,1} e^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{(s-1)/2} D_{j,2} e^{\alpha_j t}\sin(\beta_j t)\\ + \sum_{j=1}^{(s-1)/2} E_{j,1} te^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{(s-1)/2} E_{j,2} te^{\alpha_j t}\sin(\beta_j t)\end{aligned} \tag{95}$$

and for even s,

$$\begin{aligned}\phi_f(t) = C_1 + C_2 t + C_3 t^2 + C_4 e^{(-s-1)t} + C_5 te^{(-s-1)t} + \sum_{j=1}^{(s-1)/2} D_{j,1} e^{\alpha_j t}\cos(\beta_j t)\\ + \sum_{j=1}^{(s-1)/2} D_{j,2} e^{\alpha_j t}\sin(\beta_j t) + \sum_{j=1}^{(s-1)/2} E_{j,1} te^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{(s-1)/2} E_{j,2} te^{\alpha_j t}\sin(\beta_j t)\end{aligned} \tag{96}$$

Here, the constants $C_1, C_2, ..., C_5, D_{j,1}, D_{j,2}, E_{j,1}, E_{j,2}$, for all j, are arbitrary.

Note that $\phi_f(t)$ can be split into

$$\phi_f(t) = \phi_f^{homo}(t) + \phi_f^{part}(t) \tag{97}$$

where $\phi_f^{homo}(t)$ is a homogenous solution satisfying the homogeneous equation

$$g(D)\phi_f^{homo}(t) = 0 \tag{98}$$

and $\phi_f^{part}(t)$ is a particular solution for (83). When s is odd,

$$\phi_f^{homo}(t) = C_1' + \sum_{j=1}^{(s-1)/2} D_{j,1}' e^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{(s-1)/2} D_{j,2}' e^{\alpha_j t}\sin(\beta_j t) \tag{99}$$

When s is even,

$$\phi_f^{homo}(t) = C_1' + C_2' e^{(-s-1)t} + \sum_{j=1}^{s/2-1} D_{j,1}' e^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{s/2-1} D_{j,2}' e^{\alpha_j t}\sin(\beta_j t) \tag{100}$$

Here, the constants $C_1', C_2', D_{j,1}'$ and $D_{j,2}'$, for all j, are arbitrary.

Compare (99) with (95) (or (100) with (96) when s is even), there must exist a particular solution to (83) that contains the terms in (95) but not in (99) (or terms in (96) but not in (100) when s is even). Thus, we can write

$$\phi_f^{part}(t) = G_1 t + G_2 t^2 + \sum_{j=1}^{(s-1)/2} H_{j,1} t e^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{(s-1)/2} H_{j,2} t e^{\alpha_j t}\sin(\beta_j t) \tag{101}$$

and for even s,

$$\phi_f^{part}(t) = G_1 t + G_2 t^2 + G_3 t e^{(-s-1)t} + \sum_{j=1}^{(s-1)/2} H_{j,1} t e^{\alpha_j t}\cos(\beta_j t) + \sum_{j=1}^{(s-1)/2} H_{j,2} t e^{\alpha_j t}\sin(\beta_j t) \tag{102}$$

Here, we must stress that the constants $G_1, G_2, G_3, H_{j,1}, H_{j,2}$, for all j, are not arbitrary. They can be determined by plugging (101) (and (102)) into (83). We will only determine G_1 and G_2. Let us expand the polynomial $g(\lambda)$.

$$g(\lambda) = \lambda^s + b_{s-1}\lambda^{s-1} + ... + b_2\lambda^2 + b_1\lambda + b_0 - s! \tag{103}$$

From the well known Vieta's Formulas [7] that relates the coeffi cients of a polynomial with the polynomial roots, we get

$$b_0 = s! \tag{104}$$

$$b_1 = \sum_{j=1}^{s} \frac{s!}{j} = \frac{s!}{\eta} \tag{105}$$

$$b_2 = \sum_{1 \le i < j \le s} \frac{s!}{ij} \tag{106}$$

The right hand side of (83) is $s! + 2s!\phi_h(t)$, where $\phi_h(t)$ is given by (86) (or (87)). Let us consider the constant term $s! + 2s!A_1$ and the term $2s!\eta t$ on the right hand side of (83). The only way by which the left hand side of (83) can produce them is

$$(b_1 D + b_2 D^2)(G_1 t + G_2 t^2) = b_1 G_1 + 2b_2 G_2 + 2b_1 G_2 t \tag{107}$$

Hence, the following identify must hold

$$2b_1 G_2 t = 2s!\eta t \tag{108}$$

$$b_1 G_1 + 2b_2 G_2 = s! + 2s!A_1 \tag{109}$$

Substituting the expressions for b_1 from (105) and b_2 from (106), we get

$$G_1 = \eta + 2A_1\eta - 2\eta^3 \sum_{1 \le i < j \le s} \frac{1}{ij} \tag{110}$$

$$G_2 = \eta^2 \tag{111}$$

Recall that Variance$(T(x)) = f(x) - h^2(x)$. We will work in the t domain. From (86) and (87), $\phi_h(t)$ can be written as

$$\phi_h(t) = \eta t + A_1 + o(1) \tag{112}$$

Hence,

$$\phi_h^2(t) = \eta^2 t^2 + 2\eta A_1 t + A_1^2 + o(1) \tag{113}$$

From (99), (100), (101) and (102), $\phi_f(t)$ can be written as

$$\phi_f(t) = G_1 t + G_2 t^2 + A + o(1) \tag{114}$$

for some fixed A. G_1 and G_2 are given by (110) and (111). Hence,

$$\phi_f(t) - \phi_h^2(t) = (\eta - 2\eta^3 \sum_{1\le i<j\le s} \frac{1}{ij})t + A - A_1^2 + o(1) \tag{115}$$

By the following identity

$$\sum_{1\le i<j\le s} \frac{1}{ij} = \frac{1}{2}((\sum_{i=1}^{s} \frac{1}{i})^2 - \sum_{i=1}^{s} \frac{1}{i^2}) \tag{116}$$

and because $\eta = (\sum_{i=1}^{s} \frac{1}{i})^{-1}$, we get

$$\eta - 2\eta^3 \sum_{1\le i<j\le s} \frac{1}{ij} = \frac{\sum_{i=1}^{s} \frac{1}{i^2}}{(\sum_{i=1}^{s} \frac{1}{i})^3} = L(s) \tag{117}$$

Substituting t with $\ln x$, we are done with the proof. ■

We give some properties of $L(s)$ in the following lemma without a proof.

Lemma 5.9 $L(s) = \frac{\sum_{i=1}^{s} \frac{1}{i^2}}{(\sum_{i=1}^{s} \frac{1}{i})^3}$ *is a strictly decreasing function of* s*, and it tends to 0 as* $s \to \infty$*.*

Note that $\sum_i^s \frac{1}{i}$ increases like $\ln s$, as s increases. Hence, $L(s)$ decreases like $\frac{1}{\ln^3 s}$, for large s. From the numerical examples, we find that the variance of the algorithm is quite small for reasonable values of x, even when the batch size is not very large. For instance, suppose $x = e^{32}$ and $s = 64$, the variance is approximately 0.49.

6 Conclusion

In this paper, we study the statistical properties of a randomized binary search algorithm and its several variations. The algorithms are useful in load balancing or server/cache selection type of problems, and can be naturally applied in distributed and parallel settings. We give the mean and variance (exact or asymptotic) for the number of search steps in each version of the algorithm, and when possible, we give its distribution. We also analyze the access or hit pattern to the entire search space. The basic algorithm is fairly efficient in terms of the number of search steps, which also has small variance. The two batch versions of the algorithm can be used separately or combined to further reduce the number search steps and its variance.

References

[1] S.P. Bhattacharyya, H. Chapellat, and L.H. Keel. *Robust Control - The Parametric Approach*. Prentice Hall, 1995.

[2] I. S. Gradshteyn and I. M. Ryzhik. *Tables of Integrals, Series, and Products*. Academic Press, sixth edition, 2000.

[3] Abdul J. Jerri. *Introduction to Integral Equations with Applications*. Marcel Dekker, Inc., 1985.

[4] Andrei D. Polyanin and Valentin F. Zaitsev. *Handbook of Exact Solutions for Ordinary Differential Equations*. Chapman & Hall/CRC, second edition, 2003.

[5] Sheldon M. Ross. *Stochastic Processes*. John Wiley & Sons, Inc., second edition, 1996.

[6] Morris Tenenbaum and Harry Pollard. *Ordinary Differential Equations*. Dover Publications, 1985.

[7] Eric W. Weisstein. *Vieta's Formulas*. MathWorld - A Wolfram Web Resource. `http://mathworld.wolfram.com/VietasFormulas.html`.

[8] Daniel Zwillinger. *CRC Standard Mathematical Tables and Formulae*. Chapman & Hall/CRC, 31st edition, 2003.